PKI

Advanced Security

Spring 2026

Pierre-Philipp Braun <p.braun@innopolis.ru>

Table of contents

CIA triad / quadrad / polyad

- Confidentiality
- Integrity
- Availablility

bonuses:

- Non-repudiation
- Authenticity (Authentication)
- Accountability

recap from last lecture

Apply secure channels & crypto to those concepts:

…which one leverages a secret (symmetric cipher)?

…which one leverages public key cryptography?

…and which one protects against MITM?

==>

- Confidentiality –> symmetric encryption & key agreement (DH)
- Integrity –> hash function (possibly using privkey)

bonuses:

- Authenticity –> public key crypto

not sure if authentication is implicit within Confidentiality

// Questions on the tirad?

PKIX

What is the purpose of an SSL certificate?…

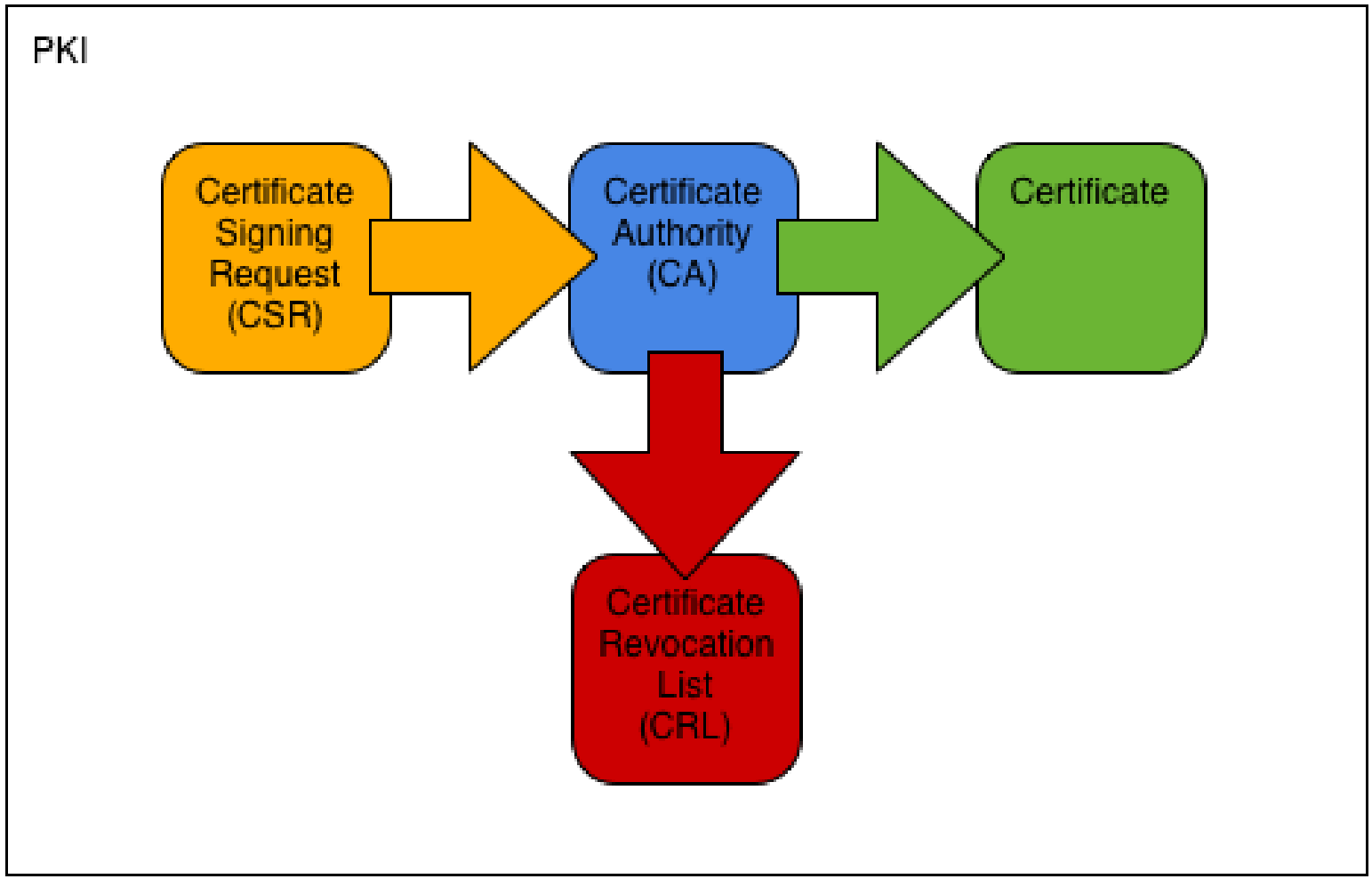
not just a key pair

==> bind pubkey & domain name **and signed by** (intermediate) **authority**


Root Certificate
Signs
Intermediate
Root Certificate
Signs
SSL Certificate

The PKIX signing process

- generate CSR (`openssl req`)
- get it signed by intermediate



Works many times… **and at the same time**

Which of those three can sign others?…

==> root CA and intermediates – just not leaf-node certificates

SSL/TLS certificate types

- Signed by official CAs (embedded Mozilla & Chrome)
- Signed by a private CA (pushed to workstations or added once)
- Self-signed (just like a root cert)

Ubuntu ships self-signed for convenience (what about Debian?)

```
/etc/nginx/snippets/snakeoil.conf
/etc/ssl/certs/ssl-cert-snakeoil.pem
/etc/ssl/private/ssl-cert-snakeoil.key
```

What is today's latest available TLS version?..

==> TLS v1.3

yes, SSL == TLS, that's just its older name

// Questions on PKIX?

Advanced SSL

What protocols are we talking about? what lives above that layer6 of ours?…

==> anything SSL/TLS and STARTTLS

* HTTPS
* S/MIME
* LDAPS
* IMAPS
* SMTP/STARTTLS
* SUBMISSION/STARTTLS
* SUBMISSIONS (465/tcp)
* ...

standards involved

- X.509
- ASN.1
- PEM vs DER
- P12 & Java key stores

PKCS #12 mainly used for storing privkey + chain-of-trust, not multiple keys

no ssl required

- when country and citizens are poor - need legacy hardware support (no https whatsoever)
- when it's a public and read-only website - why bother!
- for a package repository mirror, you might want to keep clear-text HTTP enabled
- when you're talking with a Mouse, a NetBSD power-user who runs toasters and likes to write his own clients

cannot use mixed clear-text components on a web-page that runs HTTPS

it is also possible to have dual HTTP + HTTPS for a read-only website, that is perfectly fine.

the story of Turkey a while ago

(link to be found, gov wanted citizens to install their CA…)

we have something similar here in Russia…

but let's be fair, the americans do it all the time *ipso facto*

What is HSTS?…

==> clients remember once they see that header in the HTTP response (be it http or https)

beware of the subdomains parameter! I recommend not to enable that parameter when not needed, just in case you need clear-text for some sub-domain

// Questions on advanced SSL?

Threatening SSL

What do you need to do for your own CA or self-signed cert to be accepted by the victim?…

==>

- user adds your certificate somehow
- physical access
  - linux system trust store for curl & wget
  - custom trust store for FF and Chrome
  - registry or folder path for Edge and other windows clients?
- golden road – **obtain an official certificate**

casual leaf cert: pubkey + my.domain.tld / CA: False

trick your IT dept into signing a `CA:true` for you ;-)

Let’s Encrypt

- HTTP-01 challenge
- DNS-01 challenge
- TLS-SNI-01
- TLS-ALPN-01

mhm HTTP-01 talks on port 80/tcp …

any idea there? is there anything evil we could do?…

steal from Let's Encrypt – impersonate the HTTP server

- ARP spoof front-facing IP against public GW
- ARP spoof proxied IP within the DMZ against the reverse proxy
- Take over port 80 during a very limited period of time

–> you get the challenge, you get the cert, and won't even get noticed (PoC works fine on the SNE LAN)

// Questions on SSL threats?

Other PKIs?

are there other protocols, standards and technologies that leverage PKI?…

==> DNSSEC is PKI, simply more pyramidal than PKIX

Why that?…

as pyramidal as it gets

==> chain-of-trust originates from the root zone

// Questions on other pkis?

almost done there

How come ssl works although the root certs are self-signed?…

==> *(should be clear by now)*

This is the end